Università Ca' Foscari Venezia

Computer Vision

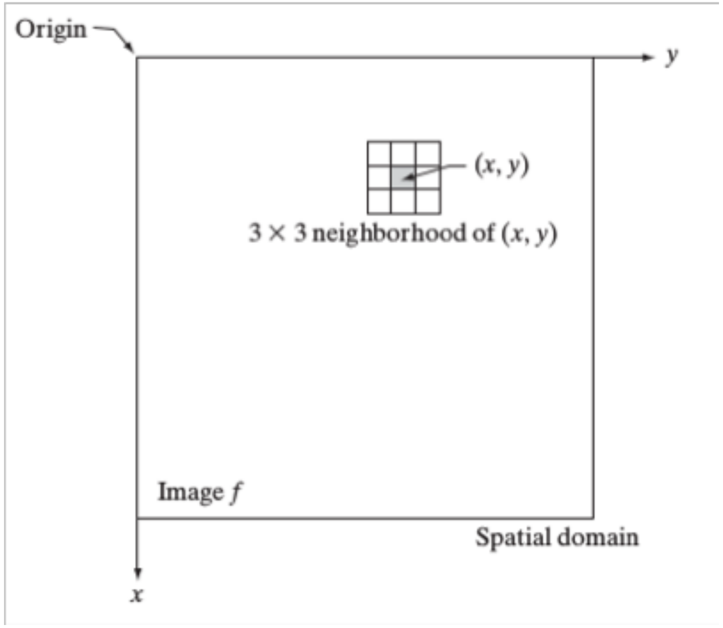
Spatial filtering

Filippo Bergamasco (filippo.bergamasco@unive.it)
http://www.dais.unive.it/~bergamasco
DAIS, Ca' Foscari University of Venice
Academic year 2018/2019

Mechanics of spatial filtering

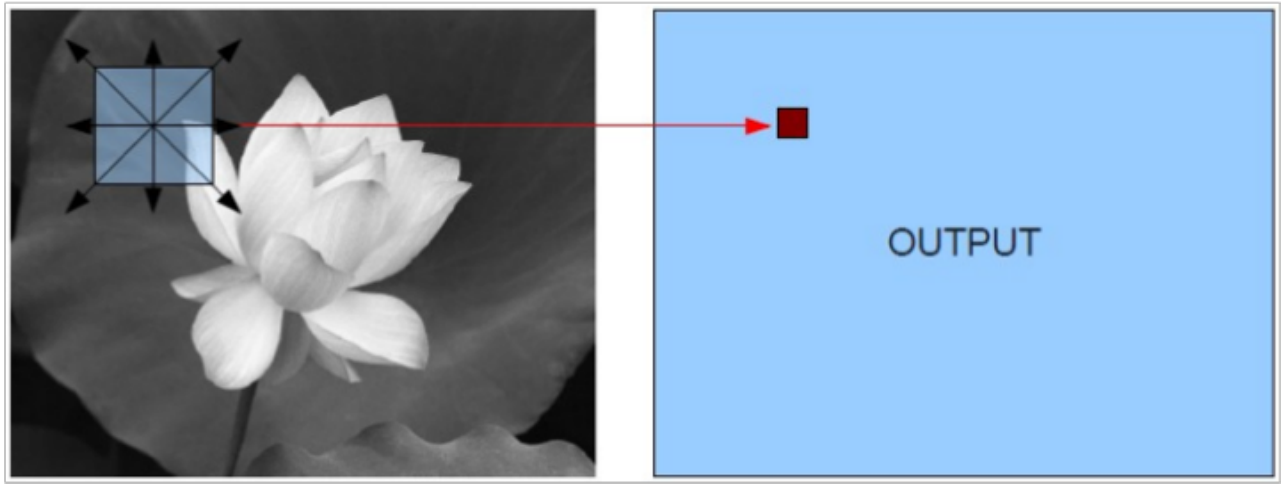
A spatial filter consists of

1. A **neighborhood** (typically a small rectangle)
2. A predefined **operation** that is performed on the image pixels encompassed by the neighborhood



Mechanics of spatial filtering

A spatial filter creates a new pixel with coordinates equal to the center of the neighborhood and whose value is the result of the filtering operation



Linear filters

Università Ca' Foscari Venezia

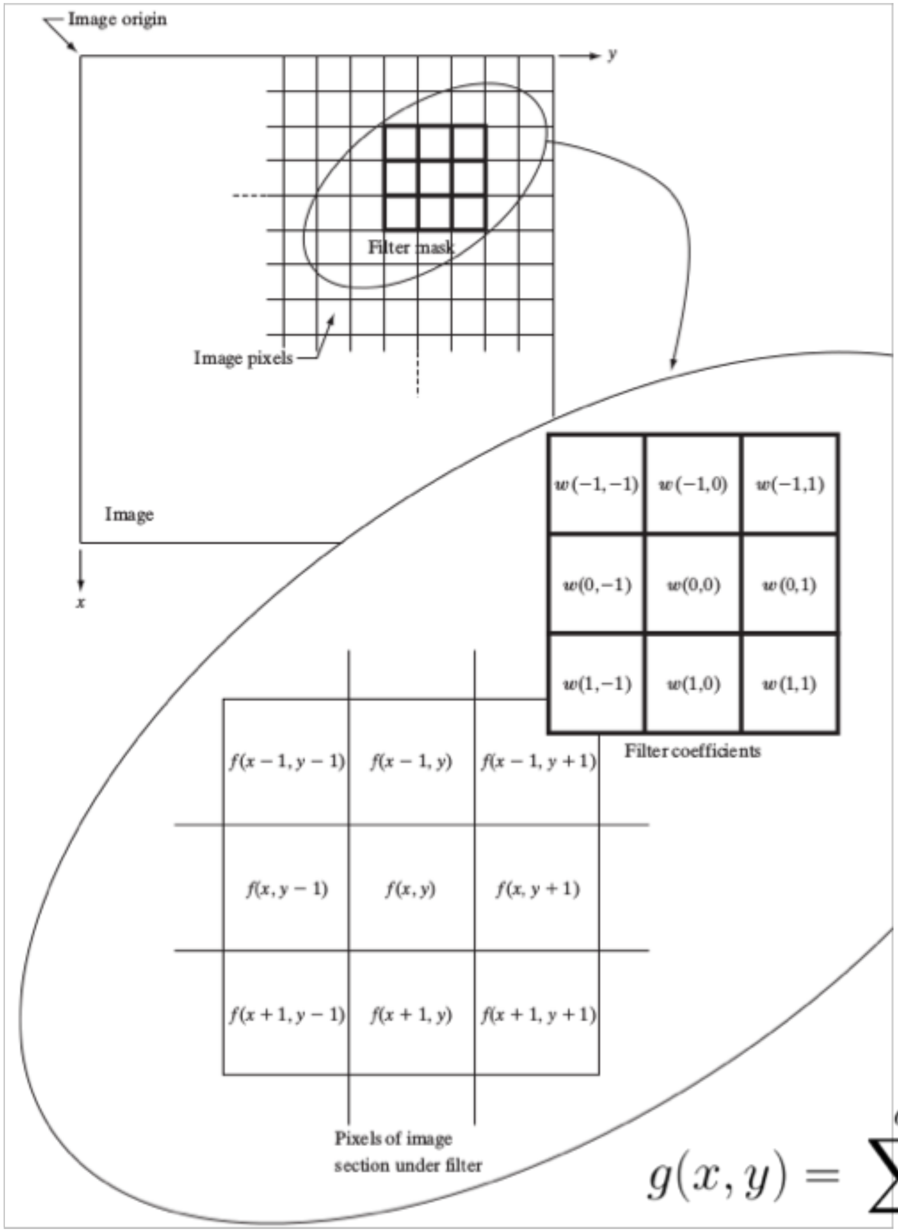
If the operation performed is linear the filter is called *linear spatial filter.*

Filter is defined in terms of a **coefficient matrix** W

The operation performed is the sum of products of the filter coefficients and the image pixels encompassed by the filter

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Linear filters



Example of a filter acting on a 3x3 neighborhood:

$$g(x,y) = w(-1,-1)f(x-1,y-1) +$$
$$+ w(-1,0)f(x-1,y) + \quad \dots$$
$$\dots$$
$$+ w(1,1)f(x+1,y+1)$$

$$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

Linear filters

Università Ca' Foscari Venezia

Observations:

- w(0,0) is aligned with the pixel at location (x,y)
- For a neighborhood of M x N we assume that M=2a+1 and N=2b+1.
  - Hence we assume filters with odd size (centered at x,y) with the smallest size being 3x3

Vector representation:

$$R = w_1 z_1 + w_2 z_2 + \dots w_{MN} z_{MN}$$

$$= \sum_{k=1}^{MN} w_k z_k$$

$$= \mathbf{w}^T \mathbf{z}$$

Filter coefficients

Image intensities encompassed by the filter

Correlation and Convolution

Linear spatial filtering can be described in terms of correlation and convolution

Correlation:

The process of moving a filter mask over a signal (the image in our case) and computing the sum of products at each location

Convolution:

Similar to correlation but the filter mask is first rotated by 180°

Università Ca' Foscari Venezia

Correlation example

Suppose that we want to compute the correlation of the 1D signal:

f(x) = 0 0 0 1 0 0 0 0

With the mask:

w(x) = 1 2 3 2 8

Correlation example

0 0 0 1 0 0 0 0

1 2 3 2 8

When moving the mask over all the possible values of the function, there are part of the two that do not overlap. The first step is to pad the function with 0 so that the filter can shift along the whole original signal

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Zero padding added to the signal

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0

After 0 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0

After 1 shift

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0

After 2 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0 8

After 3 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0 8 2

After 4 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0 8 2 3

After 5 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0 8 2 3 2

After 6 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0 8 2 3 2 1

After 7 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0 8 2 3 2 1 0

After 8 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0 8 2 3 2 1 0 0

After 9 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0 8 2 3 2 1 0 0 0

After 10 shifts

Correlation example

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

1 2 3 2 8

Result:

0 0 0 8 2 3 2 1 0 0 0 0

After 11 shifts

Correlation example

Result:

0 8 2 3 2 1 0 0

We remove the padding so that the result has the same size as the input.

Important things to notice:

- Correlation **is a function of displacement** of the filter. The first value of correlation corresponds to zero displacement, the second corresponds to one unit displacement, and so on
- Correlating a filter **w** with a function that contains all 0s and a single 1 yields a result that is a copy of w, but rotated by 180°

Università Ca' Foscari Venezia

Convolution

Convolution works exactly the same way, but the filter is rotated by 180° before the shift operations.

A fundamental property of convolution is that convolving a function with a unit impulse yields a copy of the mask at the location of the impulse

Università Ca' Foscari Venezia

2D Correlation/Convolution

In case of 2D functions, like images, the correlation works in a similar manner

For a filter of size MxN we first pad the image with a minimum of:

- M-1 rows at top and M-1 rows at bottom (filled with 0s)
- N-1 cols at left and N-1 cols at right (filled with 0s)

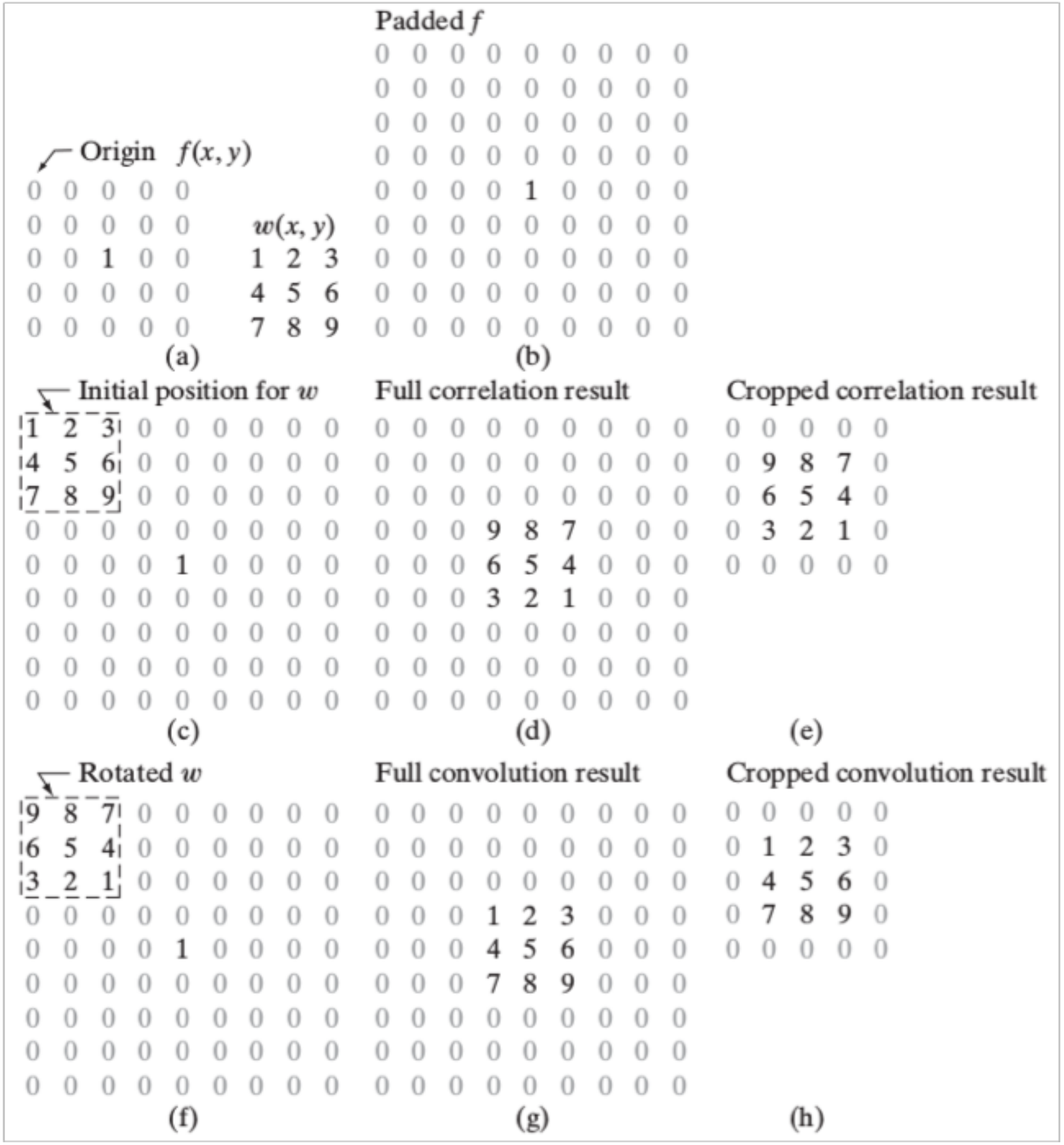
We shift the filter at each vertical and horizontal shift to perform the correlation/convolution operation:

$$(w * f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$(w \star f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Università
Ca' Foscari
Venezia

2D Correlation/Convolution



Convolution properties

Commutative

$$F \star G = G \star F$$

Associative

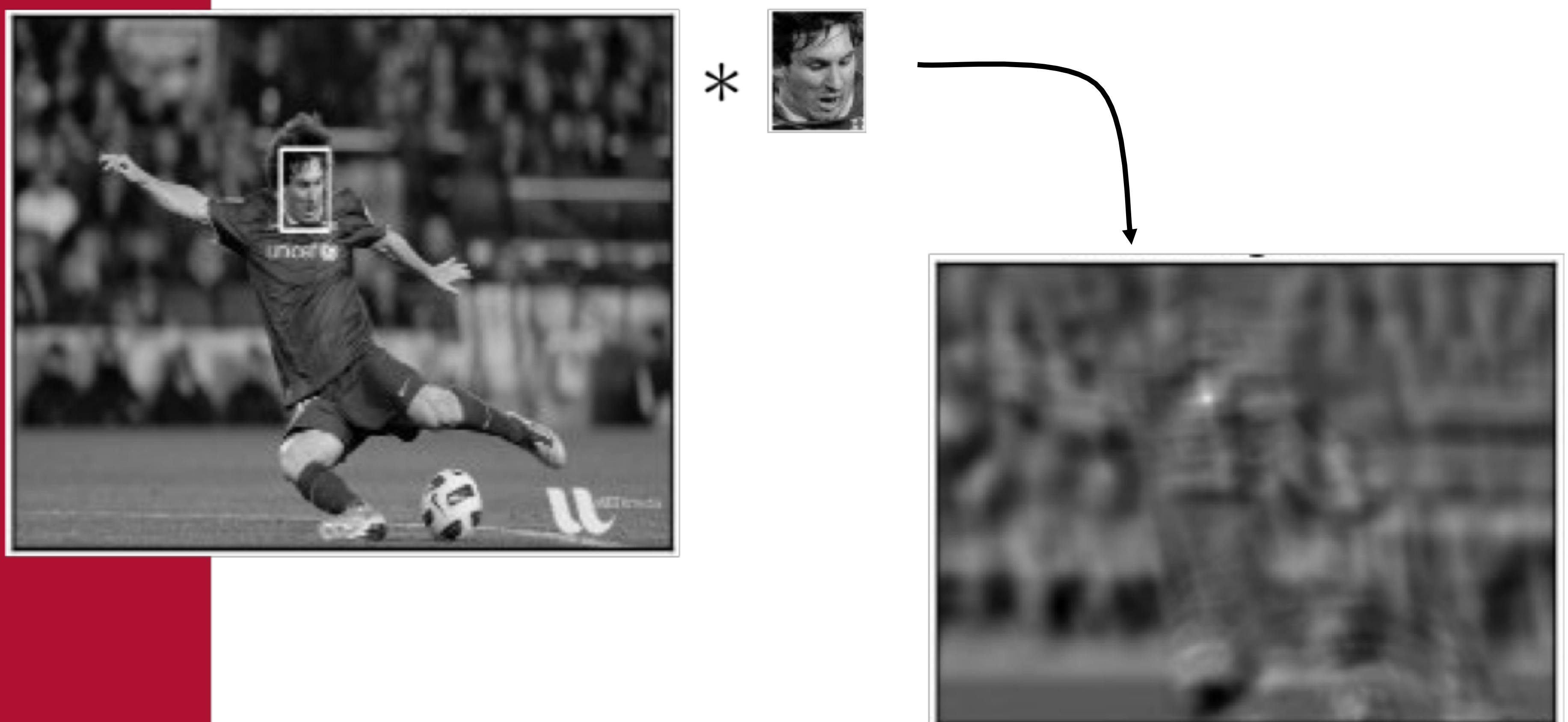
$$(F \star G) \star H = F \star (G \star H)$$

Linearity

$$(aF + bG) \star H = aF \star H + bG \star H$$

Template matching

What happens if the filter is a copy of a portion of the image?



Università
Ca' Foscari
Venezia

Smoothing spatial filters

Smoothing filters are used for blurring / noise reduction

The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.

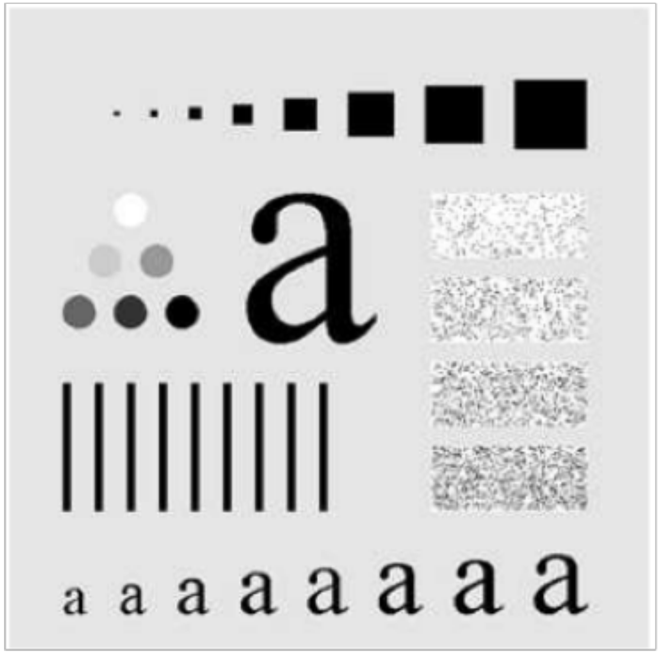
Filter masks:

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

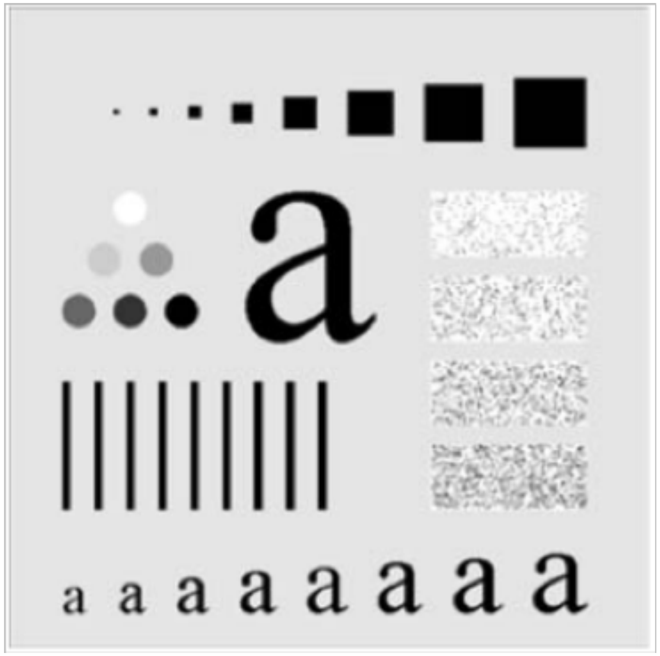
Average filter

Weighted average filter

Smoothing spatial filters

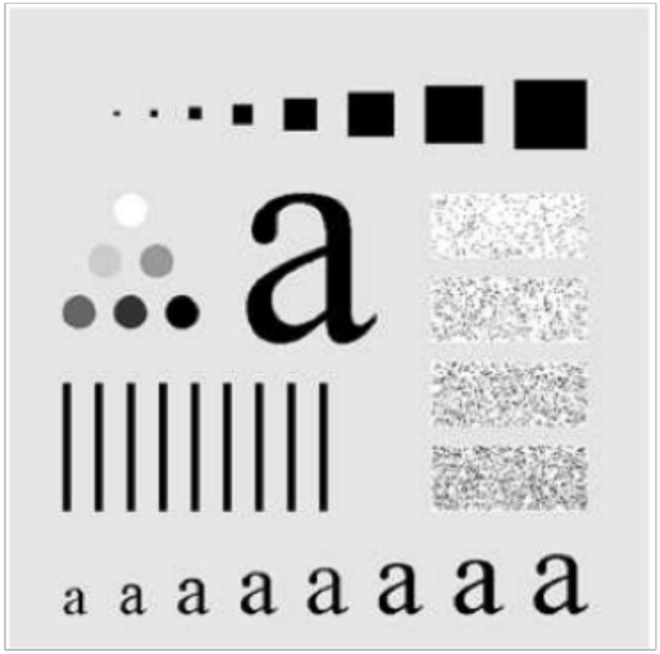


Original image



3x3 average filter

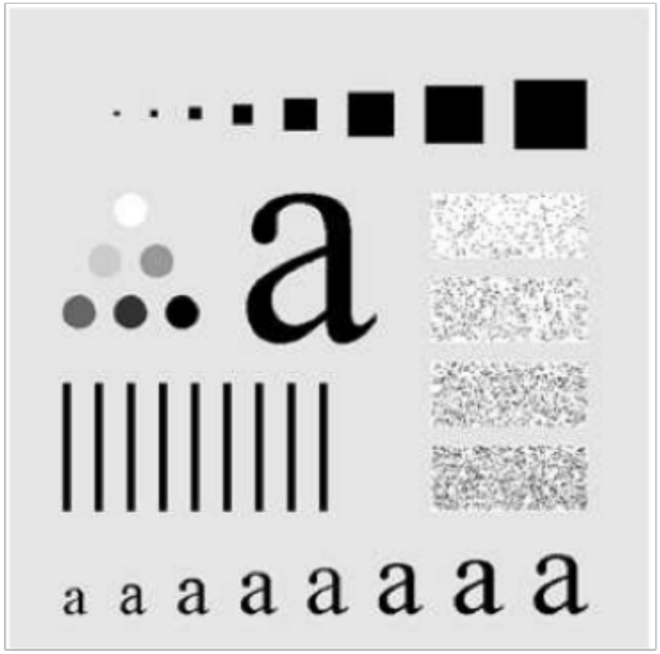
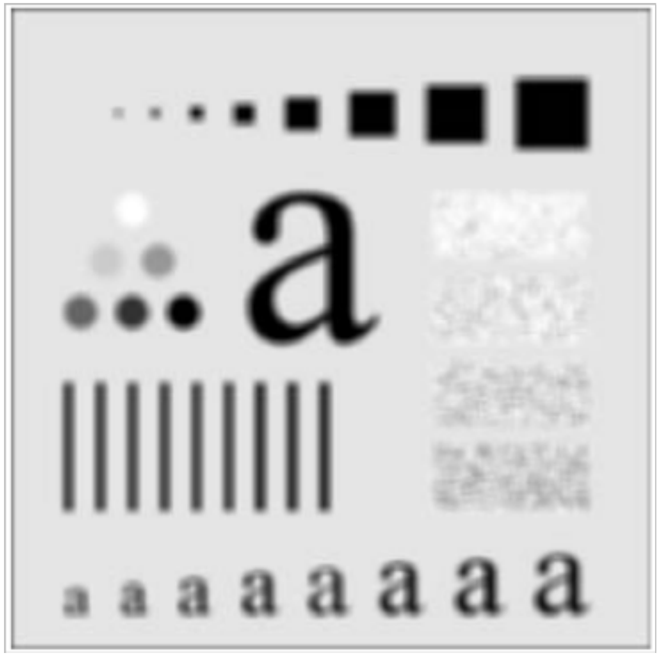
Smoothing spatial filters



Original image

5x5 average filter

Smoothing spatial filters

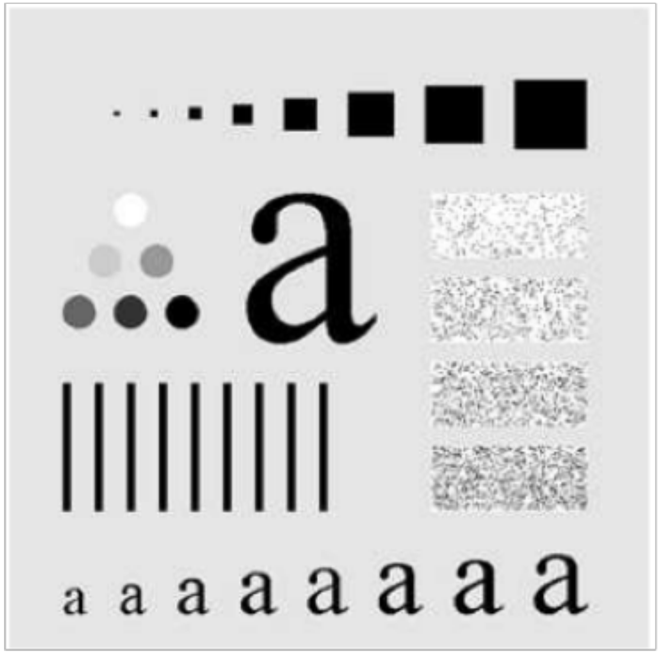



Original image

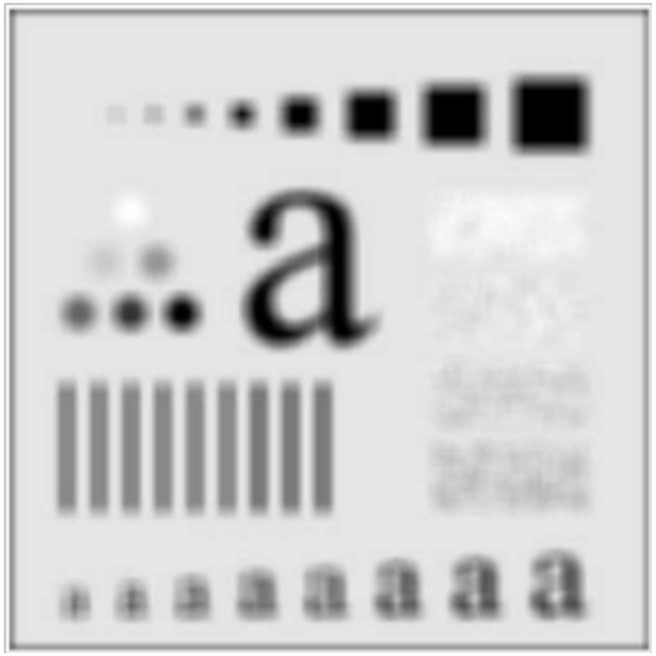
9x9 average filter

Smoothing spatial filters

Università
Ca' Foscari
Venezia



Original image



15x15 average filter

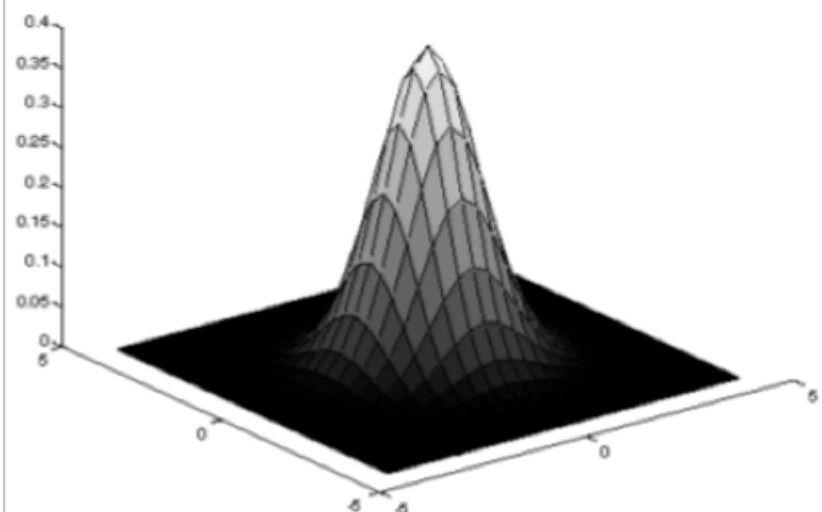
Università
Ca' Foscari
Venezia

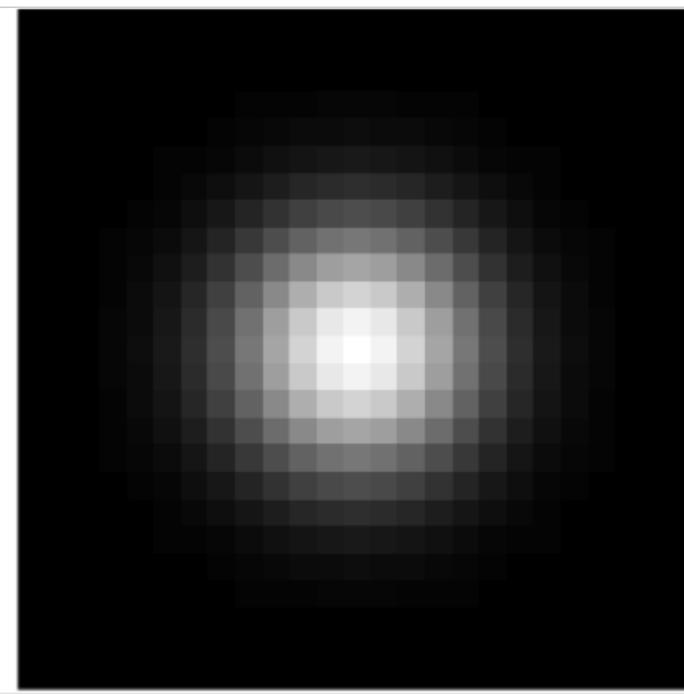
Gaussian filter

A special type of weighted average filter is the Gaussian filter.

Each element has the form

$$e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}}$$

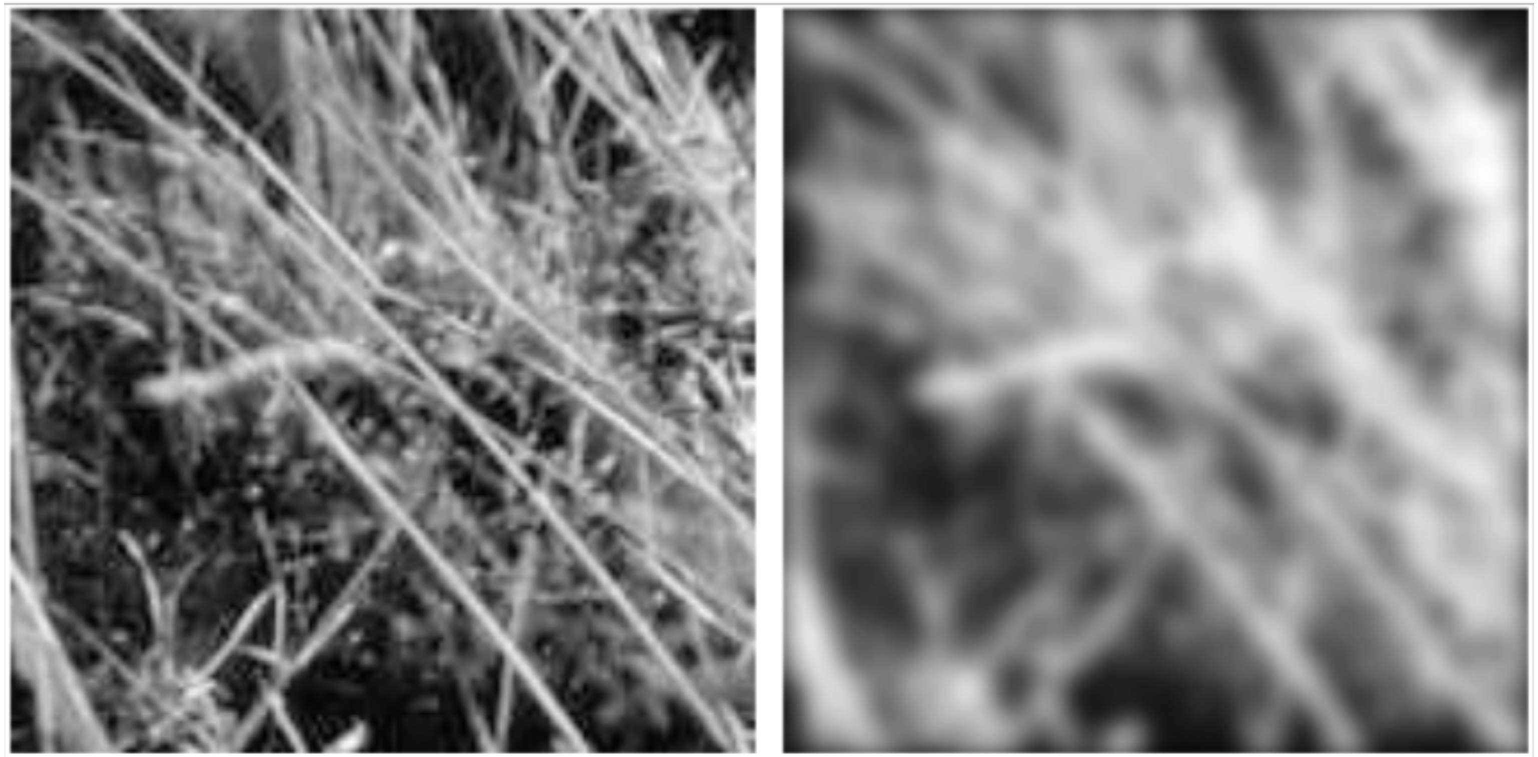




Università
Ca' Foscari
Venezia

Gaussian filter

Example of a gaussian filtered image



Order-statistic filters

Order-statistic filters are **non-linear** spatial filters whose response is based on:

1. ordering the pixels contained in the image area encompassed by the filter
2. replacing the value of the center pixel with the value determined by the ranking result.

This category of filters is non-linear and cannot be performed as a convolution/correlation

Order-statistic filters

Median-filter:

replaces the value of a pixel by the median of the intensity values in the neighbourhood of that pixel (the original value of the pixel is included in the computation of the median)

Max-filter:

Replaces the value of a pixel with the brightest intensity value in the neighbourhood

Min-filter:

Replaces the value of a pixel with the darkest intensity value in the neighbourhood

Order-statistic filters

Median-filter:

1. Sort the pixels in the MxN neighborhood of (x,y)
2. Output the (MxN)/2^th value of the sorted list

For example, in in a 3x3 neighborhood the median is the 5th largest value, in a 5x5 neighborhood it is the 13th largest value

Neighborhood values:

(10, 20, 20, 20, 15, 20, 20, 25, 100)

Sorted:

(10, 15, 20, 20, 20, 20, 20, 25, 100)

Median

Università Ca' Foscari Venezia

Filter and noise

Smoothing and median filters are particularly useful for image **denoising**

Different noise models:

1. Additive noise

$$\hat{I}(x,y) = I(x,y) + \omega$$

2. Salt & pepper (impulse) noise

$$\hat{I}(x,y) = \begin{cases} 0 \\ I(x,y) \\ L-1 \end{cases}$$

Università
Ca' Foscari
Venezia

Additive noise

Università
Ca' Foscari
Venezia

Impulse noise

Noise reduction

Which type of filter is better suited for different noise models?

Impulse noise reduction

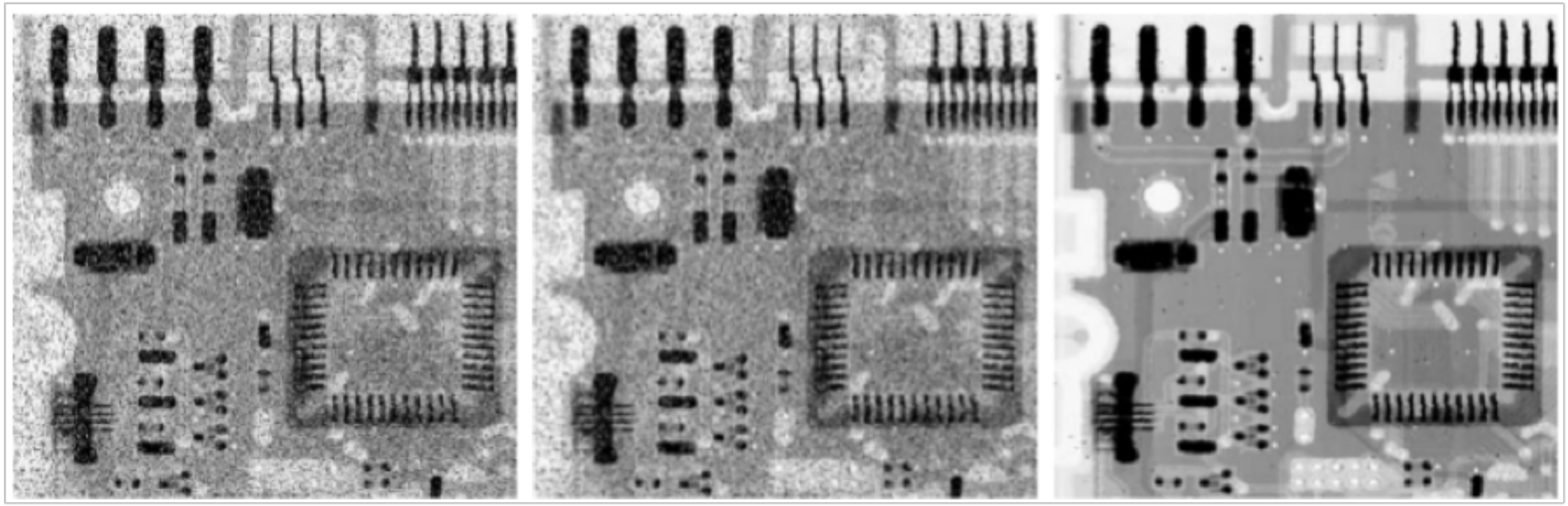


Image corrupted with impulse noise

3x3 average filter

3x3 median filter

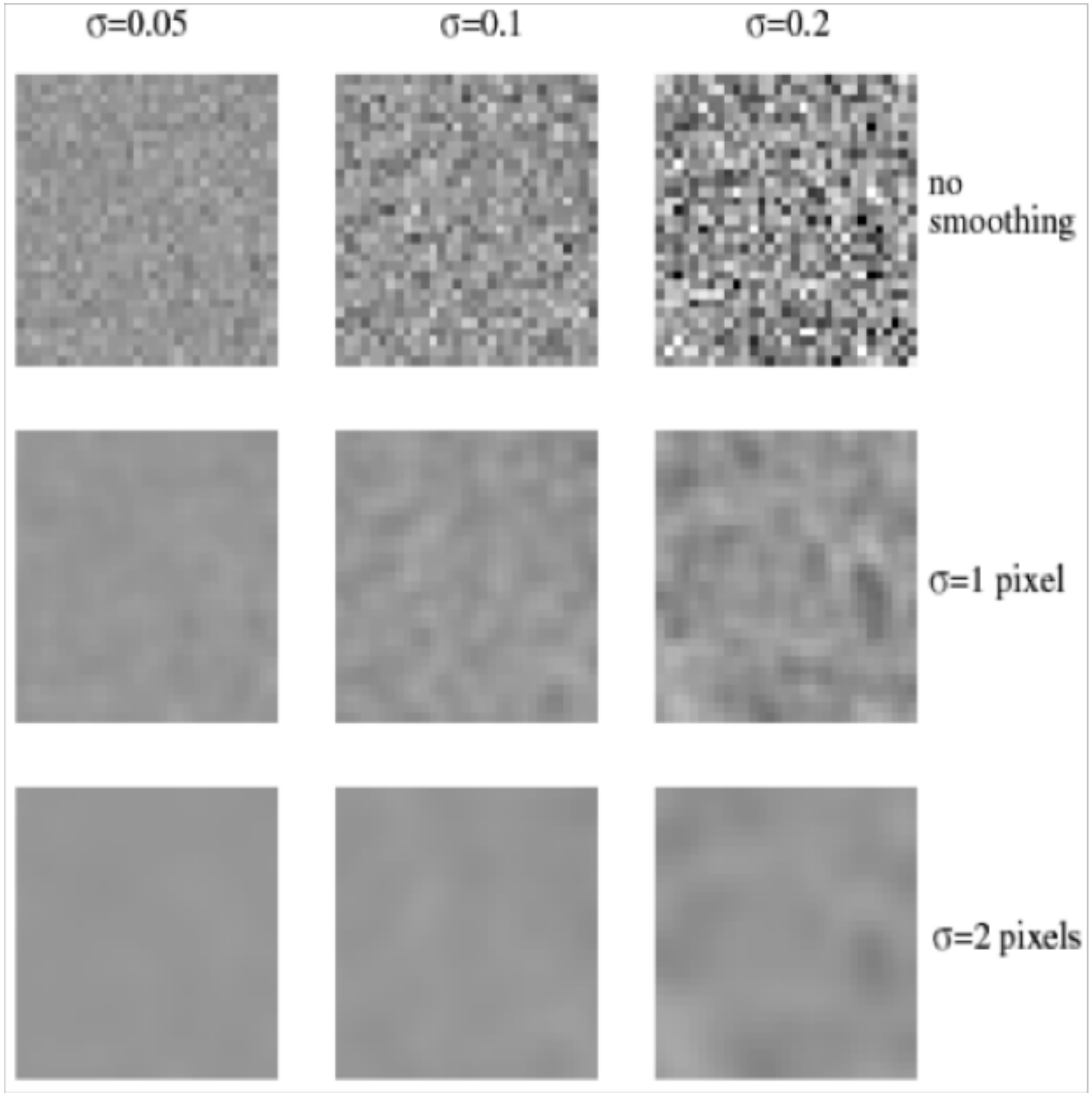
Impulse noise reduction

Image corrupted with impulse noise

3x3 average filter

3x3 median filter

Additive noise reduction



Additive noise reduction

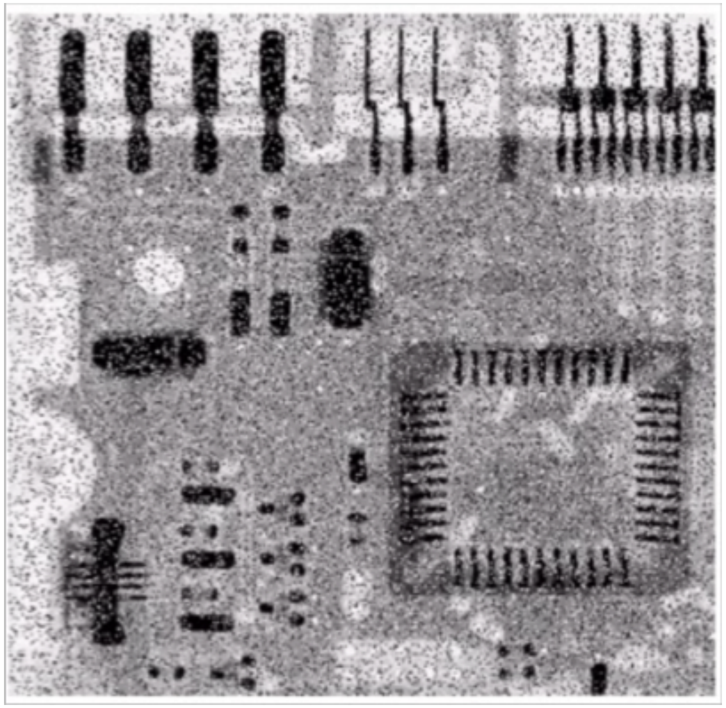
Image corrupted with
additive noise

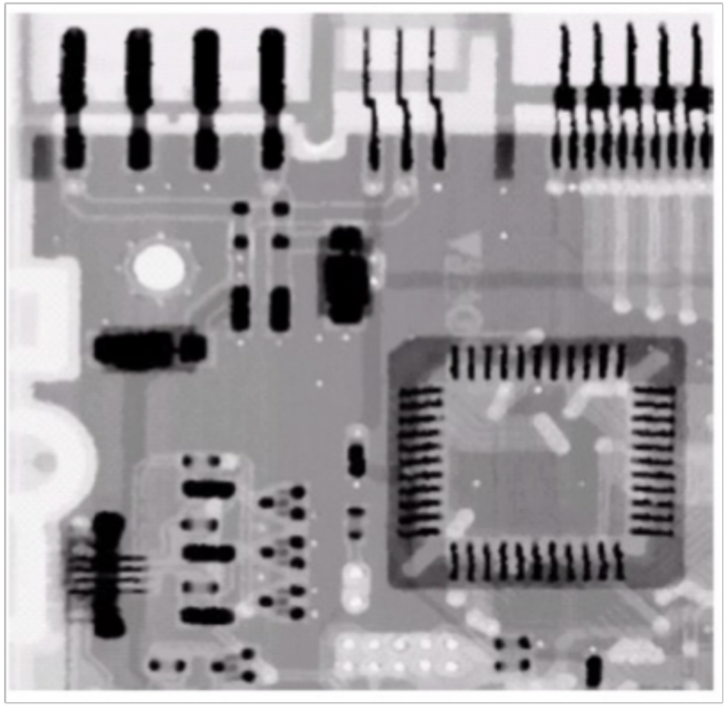
3x3 average filter

α-trimmed mean filter

To eliminate both additive and impulse noise use a robust estimate of the mean

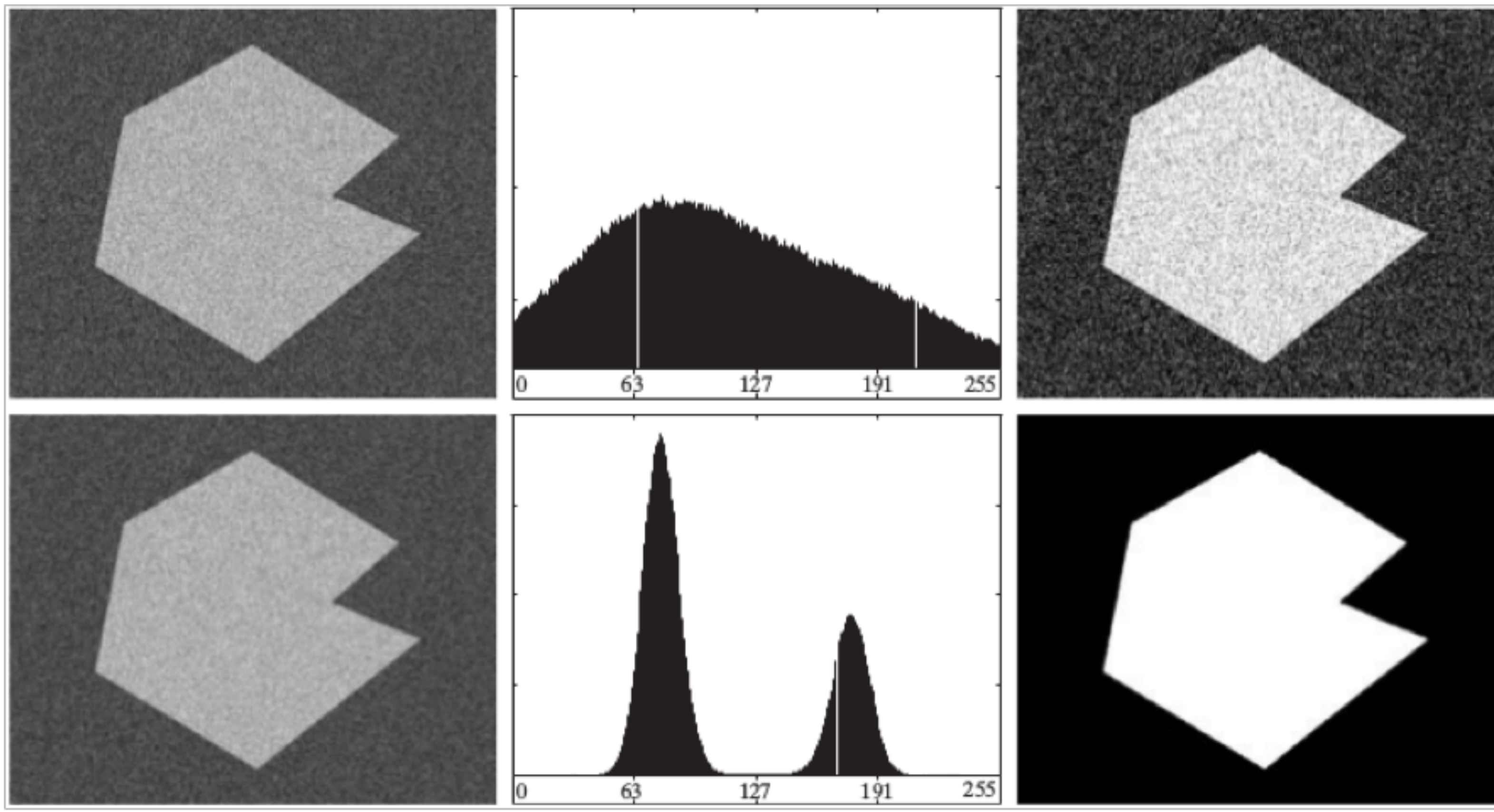
- Eliminate the top and bottom α/2 values
- Take the average of the remaining pixels





Thresholding and noise

Original noisy image | Noisy image histogram | Outsu thresholding



5x5 average filter | filtered image histogram | Otsu thresholding

Sharpening filters

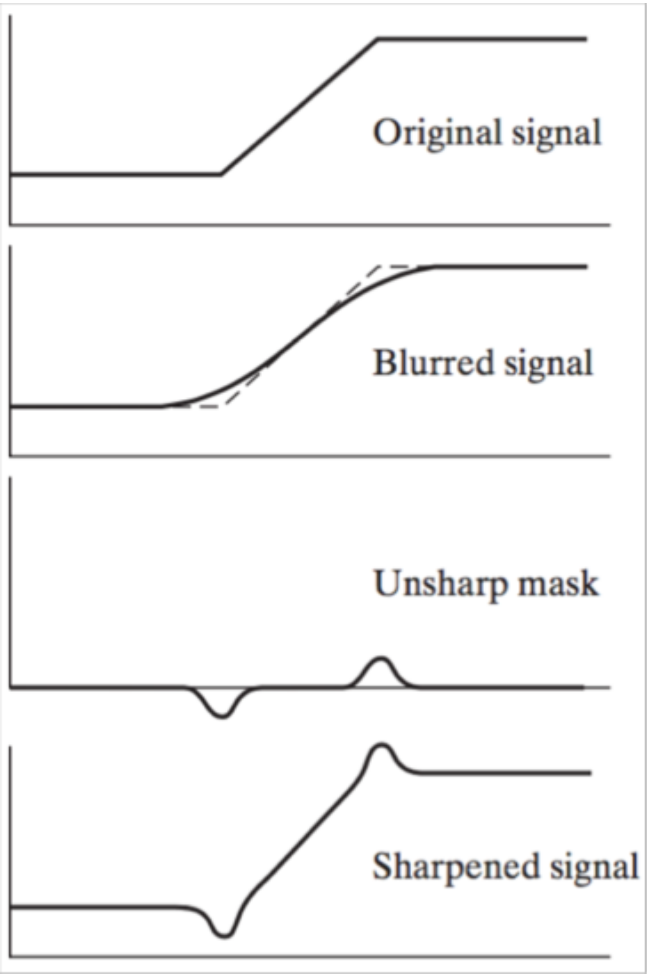
The principal objective of sharpening is to highlight transitions in intensity

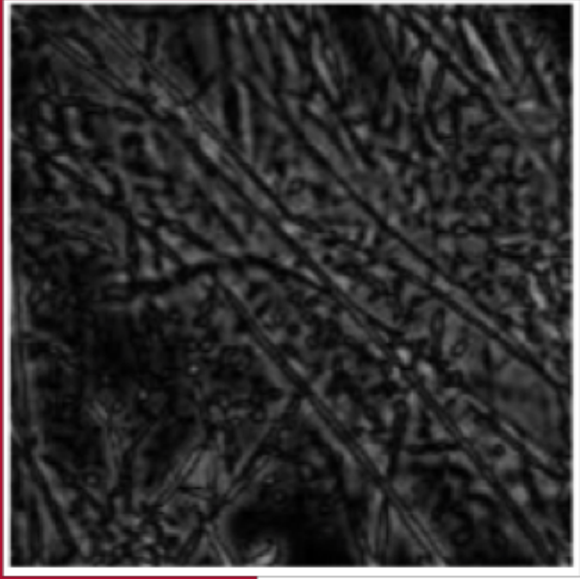
One simple approach is called **unsharp masking** (or high-boost filtering) and consist of the following:

1. Blur the original image
2. Subtract the blurred image from the the original to obtain a mask
3. Add the mask to the original (multiplied by a constant for high-boosting)

Università
Ca' Foscari
Venezia

Unsharp mask





Università Ca' Foscari Venezia

Sharpening filters

In general, sharpening filters are based on the concept of differentiation

Image blurring		Image sharpening
Averaging pixels in a neighborhood: Integration	$\longrightarrow$	Differentiation (first or second order derivatives)
Remove details		Enhance details

Università Ca' Foscari Venezia

Derivatives

Derivatives of a digital discrete functions are defined in term of differences.

First-order derivative of a one-dimensional function:

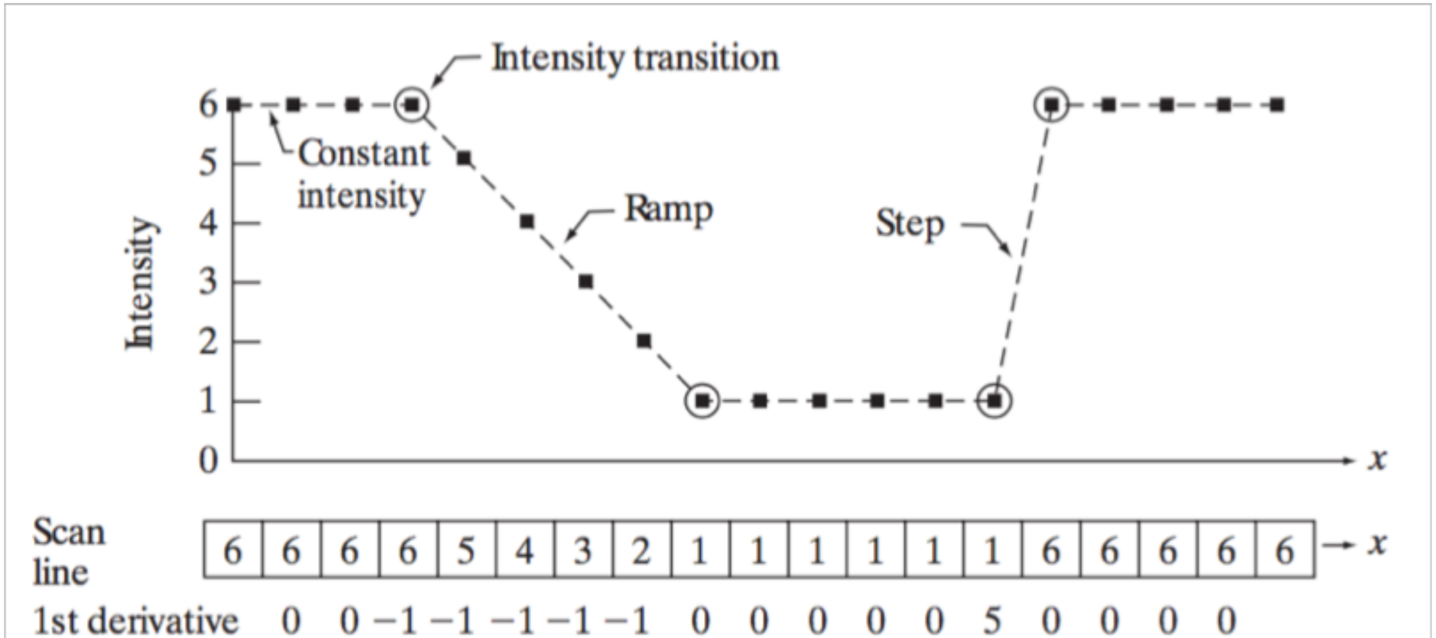
$$\frac{df}{dx} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon} \approx f(x+1) - f(x)$$

Second-order derivative:

$$\frac{d^2 f}{d^2 x} \approx f(x+1) + f(x-1) - 2f(x)$$

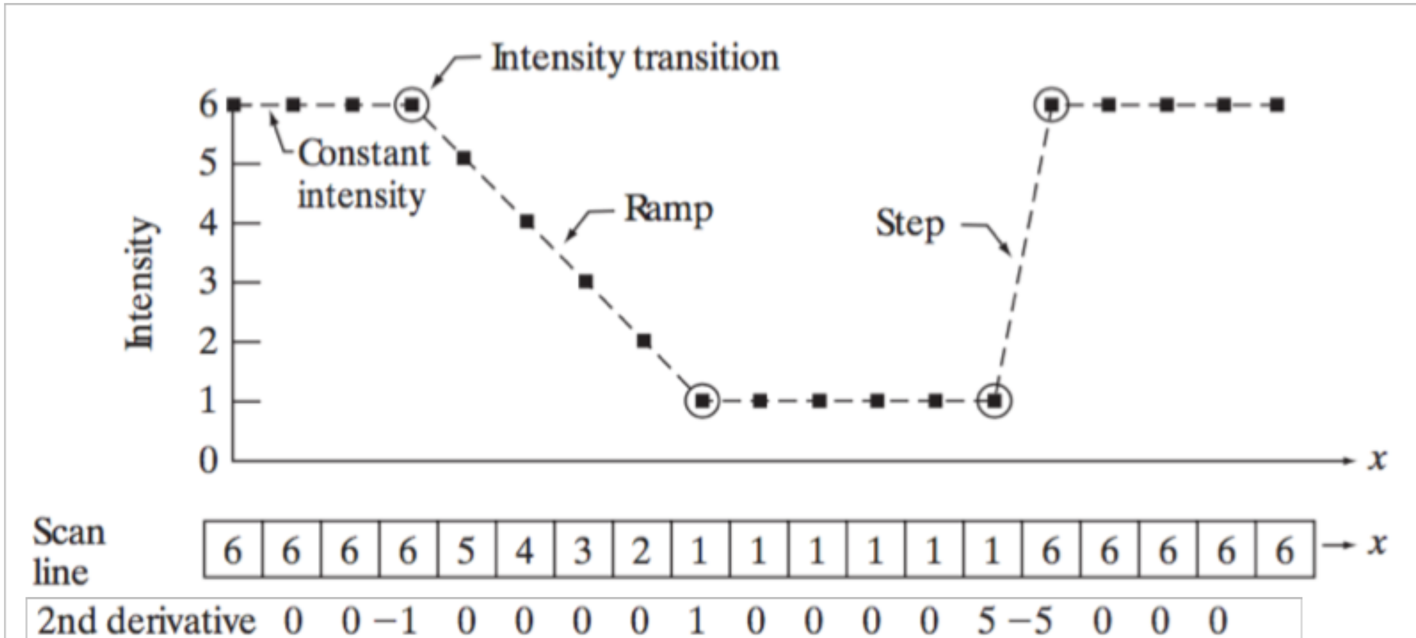
Università
Ca' Foscari
Venezia

First-order derivatives



- Zero in constant-intensity areas
- Non-zero on an intensity step or ramp
- Non-zero along ramps

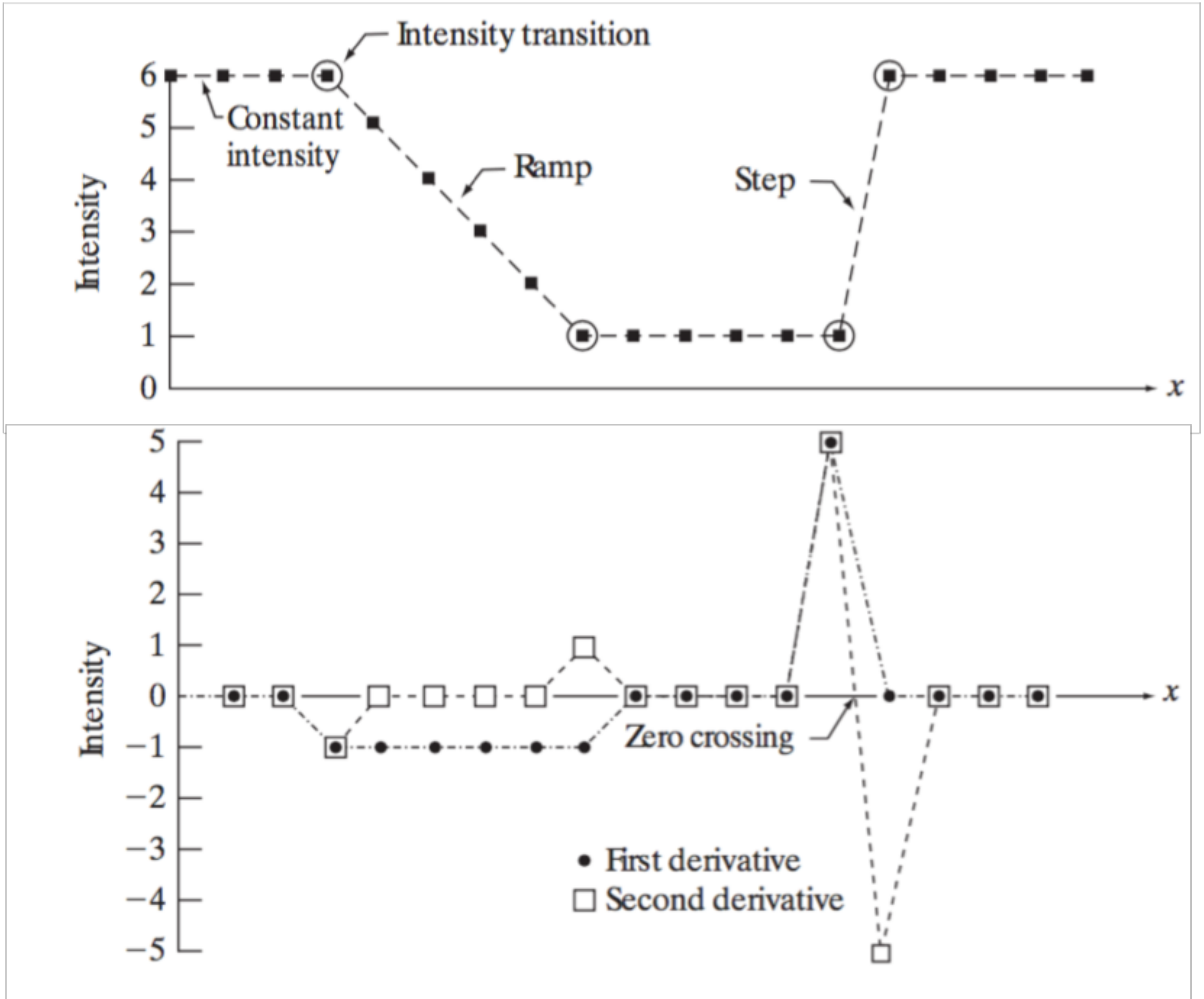
Second-order derivatives



- Zero in constant-intensity areas
- Non-zero on an onset and end of ramps and steps
- Zero along ramps

Università
Ca' Foscari
Venezia

Derivatives



Università Ca' Foscari Venezia

Sharpening

For sharpening we want to highlight intensity transitions (Edges)

First-order derivative:

would result in thick edges because the derivative is nonzero along a ramp

Second-order derivative:

would produce a double edge one pixel thick, separated by zeros

> Enhances fine detail much better than the first derivative, and are also easier to implement!

Università Ca' Foscari Venezia

Laplacian Filtering

The Laplacian is the simplest **isotropic** second-order derivative operator

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

In discrete form, the Laplacian can be expressed in term of finite differences:

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + \\ + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Laplacian Filtering

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Since derivatives are linear operators, the laplacian is a linear operator and hence can be implemented as a convolution with a proper filter mask

0	1	0
1	−4	1
0	1	0

This filter gives an isotropic result for increments of 90°

Università Ca' Foscari Venezia

Laplacian Filtering

The diagonal directions can be incorporated in the definition of the digital Laplacian by adding two more terms, one for each of the two diagonal directions.

1	1	1
1	−8	1
1	1	1

This filter gives an isotropic result for increments of 45°

Laplacian Filtering

The Laplacian operator highlights intensity discontinuities in an image and deemphasizes regions with slowly varying intensity levels

If we add the effect of the laplacian operator to the original image we are effectively sharpening the details while preserving background slowly-varying gradients

$$g(x,y) = f(x,y) + c\left(\nabla^2 f(x,y)\right)$$

Università
Ca' Foscari
Venezia

Laplacian Filtering